BS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/484,549	01/18/2000	Korbin Van Dyke	01000.9901080	9816

24228 7590 01/30/2004

MARKISON & RECKAMP, PC
PO BOX 06229
WACKER DR
CHICAGO, IL 60606-0229

EXAMINER	
ALI, SYED J	

ART UNIT	PAPER NUMBER
2127	11

DATE MAILED: 01/30/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/484,549	VAN DYKE ET AL.	
	Examiner	Art Unit	
	Syed J Ali	2127	

-- The MAILING DATE of this communication app ars on the cover she t with the correspond nce address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-12 and 14-17* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-12 and 14-17* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on December 2, 2003 has been entered.

2. This office action is in response to Amendment C, paper number 10, which was filed December 8, 2003. Claims 2-12 and 14-17 are presented for examination.

3. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Objections

4. Claim 17 is objected to because of the following informalities: In line 1, "Apparatus comprising:" should read "An apparatus comprising:". Appropriate correction is required.

Claim Rejections - 35 USC § 103

5. Claims 15-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stefaniak et al. (USPN 6,658,448) (hereinafter Stefaniak) in view of Combs et al. (USPN 6,665,701) (hereinafter Combs).

As per claim 15, Stefaniak discloses a method for providing multimedia functionality in a homogeneous multiprocessor environment comprising the steps of:

identifying available processing resources in the homogeneous multiprocessor environment (col. 5 line 47 - col. 6 line 6, "The block labeled Available Processors in Fig. 4 indicates the processors available in the system to be assigned to Affinity Group objects. Thus, all 8 processors shown in 48A through 48C are available to be assigned to Affinity Group objects 42, 43, and 44");

allocating the available processing resources among the tasks based on the capabilities of each of the available processing resources and the processing requirements of each of the tasks (col. 6 lines 7-25, "rules are specified that associate specific tasks with affinity groups", wherein the rules are defined within a specific API and assigns a task to a specific processing resource based on characteristics associated with the task and the processor affinity groupings); and

performing the tasks using the available processing resources to produce resulting data (col. 6 lines 26-47, "a thread of task X is scheduled for execution on the processors in the affinity mask of task X by the scheduler algorithm of the Windows NT operating system").

Combs discloses the following limitations not shown by Stefaniak, specifically queuing tasks (col. 5 lines 38-52, "A task-oriented resource is a resource for which requests are queued and serviced by the resource one at a time"); and

providing to the available processing resources functional programs and initial data corresponding to the tasks (col. 7 line 66 - col. 8 line 35, "The operating system call interface provides to an application program functions that the application program can invoke to read and write data to and from the hard disk, transmit data over physical data transmission networks to

remote computers, print files on a printer, and other such tasks", "The RAAU [remote access agent user] may also be implemented as a set of library functions that are linked to the application program, and therefore, together with the application program, constitute a single executable process on a computer").

It would have been obvious to one of ordinary skill in the art to combine Stefaniak with Combs since Stefaniak discloses an efficient method of identifying available processing resources and assigning tasks to those resources based on characteristics associated with those tasks and resources, but fails to provide additional scheduling flexibility. Although the resources of Combs are related more to peripheral devices than processors in a homogeneous system, the operating system functionality and scheduling technique is applicable, and provides an advantage over Stefaniak. Specifically, Combs allows tasks to be queued at a resource, allowing tasks that are not to be executed until later to be scheduled for execution. Additionally, Combs provides the benefit of being able to identify the type of task, and package together in the system call the functional programs associated with that task, such that the operating system can process the task regardless of its functionality. This would allow greater flexibility in assigning a task to a processing resource. That is, rather than scheduling a task based solely on the type of task, other factors could be taken into account and given greater importance, since the processor will have the available libraries passed to it in order to process the task.

As per claim 16, Stefaniak discloses a method for providing multimedia functionality in a homogeneous multiprocessor environment comprising the steps of:

identifying available processing resources in the homogeneous multiprocessor environment based solely on the capabilities kept track of remotely (col. 5 line 47 - col. 6 line 6, "The block labeled Available Processors in Fig. 4 indicates the processors available in the system to be assigned to Affinity Group objects. Thus, all 8 processors shown in 48A through 48C are available to be assigned to Affinity Group objects 42, 43, and 44", wherein the monitoring of resources is performed remotely, as disclosed by Combs, and discussed below);

allocating the available processing resources among the tasks (col. 6 lines 7-25, "rules are specified that associate specific tasks with affinity groups", wherein the rules are defined within a specific API and assigns a task to a specific processing resource based on characteristics associated with the task and the processor affinity groupings); and

performing the tasks using the available processing resources to produce resulting data (col. 6 lines 26-47, "a thread of task X is scheduled for execution on the processors in the affinity mask of task X by the scheduler algorithm of the Windows NT operating system").

Combs discloses the following limitations not shown by Stefaniak, specifically queuing tasks (col. 5 lines 38-52, "A task-oriented resource is a resource for which requests are queued and serviced by the resource one at a time");

keeping track, remotely from the resources, of the capabilities of all available processing resources (Abstract, "Application programs request allocation of resources from a local distributed resource allocator applications program interface. Application programs request allocation of resources from a remote distributed resource allocator system process via a resource allocator access protocol"); and

providing to the available processing resources functional programs and initial data corresponding to the tasks (col. 7 line 66 - col. 8 line 35, "The operating system call interface provides to an application program functions that the application program can invoke to read and write data to and from the hard disk, transmit data over physical data transmission networks to remote computers, print files on a printer, and other such tasks", "The RAAU [remote access agent user] may also be implemented as a set of library functions that are linked to the application program, and therefore, together with the application program, constitute a single executable process on a computer").

It would have been obvious to one of ordinary skill in the art to combine Stefaniak with Combs for reasons discussed above in reference to claim 15.

6. Claims 2-3, 5-12, 14, and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stefaniak in view of Combs in view of Borrill (USPN 6,496,922).

As per claim 2, Borrill discloses the following limitations not shown by the modified Stefaniak, specifically the method of claim 15 wherein a plurality of processors of the homogeneous multiprocessor environment are capable of executing a first instruction of a first instruction set and a second instruction of a second instruction set (col. 3 lines 46-62, "The current invention is described herein in an embodiment for statelessly executing instructions written for multiple, different instruction set architectures on a single platform").

It would have been obvious to one of ordinary skill in the art to combine the modified Stefaniak with Borrill since the modified Stefaniak allows a task to be bundled with its

functional programs such that any processor would have the available libraries to process it, but does not take into account system calls that originate from different types of systems. With the increasing development of different types of instruction set architectures, a need exists to create compatibility between platforms. Borrill provides a way of emulating different instruction set architectures through conversion and look-up tables, such that any type of task from any type of instruction set architecture can be executed on a single hardware platform.

As per claim 3, Borrill discloses the method of claim 2 wherein the first instruction and the second instruction share an identical bit pattern but perform different operations (col. 4 lines 5-14, "Three bits will identify eight different instruction set architectures, so the number of bits for the tags T0-T8 [corresponding to instructions 70A-78A, respectively] may be varied depending upon how many instruction set architectures are to be accommodated", wherein the same bit pattern may be interpreted differently based on the three bit long identification code that identifies the type of instruction).

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 5, Borrill discloses the method of claim 3 further comprising the step of:

converting a functional program of the functional programs expressed using the first instruction set to an equivalent functional program expressed using the second instruction set (col. 5 lines 19-29, "The function of the DDU's is to convert the incoming instructions into instructions that are recognizable by the native processor 90. This involves translating

instructions one-to-one from the non-native ISA to the native ISA [which can be done by a look-up table]; converting complex instructions into several native ISA instructions [e.g., complex x86 instructions may translate into several VLIW instructions]; implementing condition codes of the non-native ISA as native-ISA condition codes; and so on").

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 6, Borrill discloses the method of claim 3 wherein the tasks comprise:

x86 processing;

graphic image processing;

video processing;

audio processing; and

communication processing (col. 3 lines 46-62, "It is equally applicable to the execution of any set of instructions, even different sets of different original sizes, on the single platform using instructions of greater length than the original, non-native instruction set architecture[s]", wherein any type of instruction set may be supported and provided with a translation such that the instructions may be executed on any available processor).

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 7, Combs discloses the method of claim 3 further comprising the step of:

receiving the initial data from a first input/output device (col. 7 lines 46-65, "The computer 601 comprises...one or more input/output controllers 605, and a network controller 606. These various components exchange data over at least one internal bus 607", wherein the input/output device passes data to the CPU for processing in accordance with the RAAU operating system call to provide the data needed to perform processing for that device).

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 8, Combs discloses the method of claim 3 further comprising the step of:

passing the resulting data to a first input/output device (col. 7 lines 46-65, "The computer 601 comprises...one or more input/output controllers 605, and a network controller 606. These various components exchange data over at least one internal bus 607", wherein communication occurs over the internal bus for the purpose of passing requests to the CPU as well as returning results to the input/output device via the communications bus and input/output controller).

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 9, Combs discloses the method of claim 8 wherein the step of passing the resulting data to the first input/output device further comprises the step of:

passing the resulting data through an intermediary device, wherein the intermediary device is coupled to the first input/output device and to a second input/output device (col. 7 lines 46-65, "The computer 601 comprises...one or more input/output controllers 605, and a network

controller 606. These various components exchange data over at least one internal bus 607", wherein the communications bus and the input/output controllers are intermediary devices that coordinate the processing of requests and data associated with the operating system calls).

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 10, Combs discloses the method of claim 9 wherein the step of passing the resulting data through an intermediary device, wherein the intermediary device is coupled to the first input/output device and to a second input/output device further comprises the step of:

automatically adapting to a reallocation of the available processing resources among the tasks (col. 3 lines 9-35, "Each resource allocator system agent maintains a database of global network resource information and constantly communicates with all other resource allocator system agents that comprise the distributed resource allocator handling system to ensure that each resource allocator system agent has the same global network information").

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 11, Combs discloses the method of claim 8 wherein the step of passing the resulting data to a first input/output device further comprises the step of:

passing the resulting data to a mixed-signal device (Fig. 1 elements 605, 606, and 607, wherein it is interpreted that a mixed-signal device is any device capable of digital to analog conversion or vice versa, and the devices described above may serve as intermediary devices,

such as a routers and switches, and are capable of acting as such a digital to analog converter or analog to digital converter).

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 12, Stefaniak, as modified by Combs, discloses the method of claim 3 wherein the step of allocating the available processing resources among the tasks is dynamically adjusted (col. 5 line 47 - col. 6 line 6, "The block labeled Available Processors in Fig. 4 indicates the processors available in the system to be assigned to Affinity Group objects. Thus, all 8 processors shown in 48A through 48C are available to be assigned to Affinity Group objects 42, 43, and 44"; col. 6 lines 7-25, "rules are specified that associate specific tasks with affinity groups", wherein Combs discloses monitoring the available global resources and changing the list as resources become available, and when taken in combination with Stefaniak would suggest reassigning tasks if processing resources become available and/or unavailable).

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 17, Stefaniak discloses [an a]pparatus comprising:

a plurality of processors coupled to a bus (Fig. 2 element 26).

Combs discloses the following limitations not shown by Stefaniak, specifically an input/output interface coupled to the bus (Fig. 6 element 605).

Stefaniak, as modified by Combs, discloses a plurality of input/output devices coupled to the input/output interface, the plurality of processors processing program code configured to perform a plurality of tasks, the program code comprising:

program code configured to cause a first portion of the plurality of processors to interact with a first input/output device of the plurality of input/output devices (col. 6 lines 7-25, "rules are specified that associate specific tasks with affinity groups", wherein the rules are customizable for association of processors with tasks and/or devices, such that processors are divided into groups and allocated accordingly); and

program code configured to cause a second portion of the plurality of processors to interact with a second input/output device of the plurality of input/output devices (col. 6 lines 7-25, "rules are specified that associate specific tasks with affinity groups", wherein the rules are customizable for association of processors with tasks and/or devices, such that processors are divided into groups and allocated accordingly);

wherein the first portion of the plurality of processors provide functionality as found in a first application-specific subsystem and wherein the first input/output device is the first application-specific subsystem (col. 3 lines 1-5, "An object of this invention is to provide both a user-interface and an application programming interface that give system administrators the ability to associate NT operating system tasks with specific processor groups, which are in turn associated with specific CPUs", wherein the processing rules are definable such that tasks associated with a specific input/output device are routed to the affinity group that is best suited to process the application tasks for that device); and

wherein the second portion of the plurality of processors provide functionality as found in a second application-specific subsystem and wherein the second input/output device is the second application-specific subsystem (col. 3 lines 1-5, "An object of this invention is to provide both a user-interface and an application programming interface that give system administrators the ability to associate NT operating system tasks with specific processor groups, which are in turn associated with specific CPUs", wherein the processing rules are definable such that tasks associated with a specific input/output device are routed to the affinity group that is best suited to process the application tasks for that device).

Borrill discloses the following limitations not shown by Stefaniak or Combs, specifically program code configured to cause a second portion of the plurality of processors to emulate a specific microprocessor instruction set (col. 3 lines 46-62, "The current invention is described herein in an embodiment for statelessly executing instructions written for multiple, different instruction set architectures on a single platform").

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

As per claim 14, Stefaniak, as modified by Combs, discloses the apparatus of claim 17 further comprising:

kernel program code configured to dynamically allocate the processing of the program code among the plurality of processors (col. 5 line 47 - col. 6 line 6, "The block labeled Available Processors in Fig. 4 indicates the processors available in the system to be assigned to Affinity Group objects. Thus, all 8 processors shown in 48A through 48C are available to be

assigned to Affinity Group objects 42, 43, and 44"; col. 6 lines 7-25, "rules are specified that associate specific tasks with affinity groups", wherein Combs discloses monitoring the available global resources and changing the list as resources become available, and when taken in combination with Stefaniak would suggest reassigning tasks if processing resources become available and/or unavailable).

It would have been obvious to one of ordinary skill in the art to combine Stefaniak, Combs, and with Borrill for reasons discussed above in reference to claims 15 and 2.

7. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Stefaniak in view of Combs in view of Borrill in view of Guyer et al. (USPN 4,597,041) (hereinafter Guyer).

As per claim 4, Guyer discloses the following limitations not shown by the modified Stefaniak, specifically the method of claim 3 wherein a first processor of the plurality of processors executes an input/output kernel program, the input/output kernel program including a first portion expressed using the first instruction set and a second portion expressed using the second instruction set (col. 17 lines 25-37, "kernel microcode includes at least a portion of the NOVA instruction set microcode and is responsive to single character commands provided from a terminal through SIO 210. Vertical microcode include microcode for the full NOVA, ECLIPSE, and MV/8000 instruction sets and is responsive to multiple character commands provided from a terminal").

It would have been obvious to one of ordinary skill in the art to combine the modified Stefaniak with Guyer since the availability of multiple instruction sets increases the capabilities

of a system. Specifically, executing a kernel such that it is compatible with multiple instruction sets allows the system to interact with many different types of devices, and eliminates certain hardware needs. For instance, a system with devices implemented in different instruction sets would normally require a specific processor for each instruction set. By allowing a single processor to execute instructions of different instruction sets, the need for an additional processor is eliminated, increasing the efficiency of that system.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Syed Ali
January 15, 2004

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100